

СИСТЕМА УЧЕТА ПОСЕЩАЕМОСТИ СТУДЕНТОВ

А.О. Банщиков, А.А. Гируцкая, И.В. Цапко
Томский политехнический университет
aob19@tpu.ru

Введение

Основную часть информации о внешнем мире человек получает по зрительному каналу и далее весьма эффективно обрабатывает полученную информацию при помощи аппарата анализа и интерпретации визуальной информации. Поэтому встает вопрос о возможности машинной реализации данного процесса. За счет возрастания сложности решаемых научно-технических задач, автоматическая обработка и анализ визуальной информации становятся все более актуальными вопросами.

Поэтому целью нашей работы являлось написание программного обеспечения, которое предоставило бы пользователям возможность поиска человека в видеопотоке и установления его личности. Данные технологии могут использоваться в повседневных ситуациях. Мы остановились на контроле посещаемости студентами занятий в университете.

Описание алгоритма

Для решения задачи поиска человека в видеопотоке было разработано приложение в среде Py-Charm на языке Python. Поиск человека производится с использованием детектора Хаара с использованием сверточной нейронной сети ResNet34.

Для обнаружения лиц был использован детектор Хаара. Его основным преимуществом является скорость. Благодаря быстрой обработке изображения, можно с лёгкостью обрабатывать потоковое видео. Детектор Хаара используется для распознавания большинства классов объектов. К ним относятся лица и другие части тела людей, номера автомобилей, пешеходы, дорожные знаки, животные и т.д. Детектор Хаара реализован в библиотеке OpenCV.

Для распознавания человека на фотографии была использована библиотека машинного обучения dlib, которая содержит удобные средства распознавания лиц.

Для решения задачи верификации была использована сверточная нейронная сеть, а именно предварительно обученная нейронная сеть ResNet. От сети отрезаются слои, отвечающие за классификацию, и остаются только сверточные слои, которые извлекают ключевые признаки из изображения. Результат работы - набор чисел, который называется дескриптором. Такие дескрипторы мы извлечем из фотографий и с web-камеры.

dlib использует модифицированный вариант сети ResNet34. Эта сеть выдает дескрипторы из 128 чисел. Сеть обучена специальным образом так, чтобы дескрипторы фотографий одного человека находились рядом друг с другом, а дескрипторы фотографий разных людей - далеко друг от друга.

dlib использует аффинное преобразование изображения с использованием ключевых точек на лице. Производится перенос ключевых точек в такую позицию, как будто человек смотрит прямо в камеру. Дескрипторы извлекаются только после аффинного преобразования изображения.

Чтобы оценить близость дескрипторов в dlib используется Евклидово расстояние. Если значение Евклидова расстояния между дескрипторами меньше 0.6, то считается, что на изображениях один и тот же человек. С использованием такой метрики dlib обеспечивает точность 99.38% на тесте распознавания лиц Labeled Faces in the Wild.

Результаты работы

Для оценки алгоритма были проведены испытания программы. В процессе тестирования, были обработаны фотографии студентов, обучающихся в группе 8И5А. Полученные дескрипторы были добавлены в документ формата JSON.

Далее был использован захват видеопотока с камеры, где происходит обнаружение лиц в кадре. Каждое лицо выделяется рамкой прямоугольной формы. Затем из этой области производится получение дескриптора и его сравнение с имеющимися дескрипторами. Если запись о человеке существует, то система фиксирует время прибытия студента на занятие. Система успешно распознает и идентифицирует людей, находящихся в движении, а также может распознать несколько человек, одновременно находящихся в кадре.

Таким образом, если студент обнаружен впервые, то фиксируется лишь время прибытия студента на занятие, время ухода не указывается. При каждом повторном обнаружении, записывается время ухода. Если студент был замечен более двух раз, то временем ухода считается последнее появление студента в кадре.

На рисунке 1 представлена процедура снятия дескриптора с изображения. При использовании фотографии с высоким качеством, вероятность верной идентификации человека в видеопотоке выше.

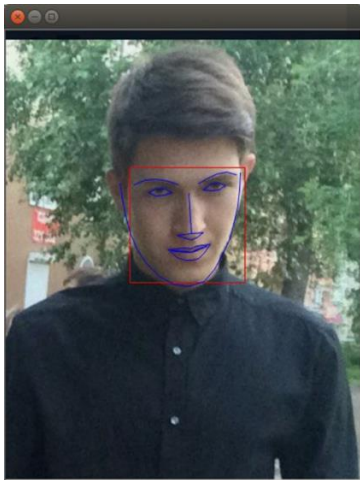


Рис. 1. Получение формы лица и снятие дескриптора с изображения

С целью ускорения работы программы, обновление данных в документе производится не при каждом обнаружении человека в видеопотоке, а с заданным интервалом времени. Данные помещаются в динамическую структуру, с указанием личности человека и временем его обнаружения. Затем, по истечении времени, производится перезапись документа JSON актуальными данными.

На рисунке 2 представлена работа приложения.

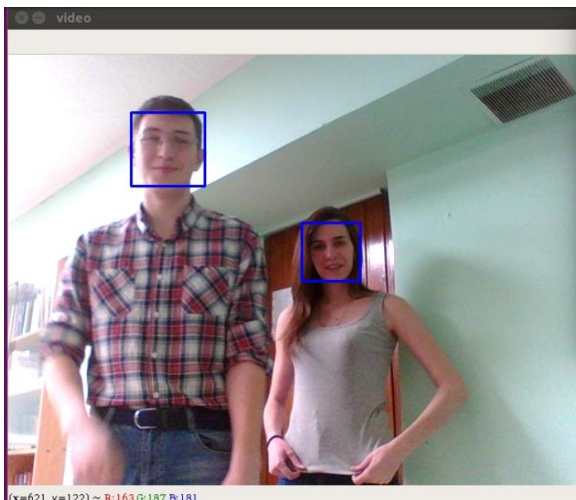


Рис. 2. Распознавание людей в реальном времени

В программе имеется возможность формирования отчета по посещаемости занятия студентами (рис. 3). Это происходит по инициативе пользователя, путем нажатия на кнопку.

Отчет		
Студент	Время прихода	Время ухода
Girutskaya Anna	12:14	12:30
Banshchikov Alexander	12:14	12:32
Potapkina Serafima	12:15	12:22
Pravosudov Matvey	12:15	12:21
Mirtov Sergey	12:16	12:23
Demidenko Ludmila	12:17	12:22
Sokolova Evgenia	12:17	12:22
Botkina Julia	12:18	12:21
Sergeeva Anastasia	12:19	12:27
Margulis Stanislav	12:21	12:26
Ngok Bick	12:24	12:26

Рис. 3. Пример формы отчета

Заключение

Основываясь на результатах тестирования приложения, можно сделать вывод о том, что алгоритм распознавания человека в видеопотоке является работоспособным и определяет человека с заявленной точностью.

Решение может быть использовано в университетах, школах, различных предприятиях для контроля посещаемости, фиксации времени прибытия людей и их ухода.

В ходе тестирования были выявлены некоторые проблемы, а именно:

1. Качество исходных фотографий существенно влияет на работу алгоритма распознавания человека;
2. Алгоритм не смог различить изображения близнецов;

Список использованных источников

1. Обучение каскада Хаара на примере поиска символов автомобильного номера OpenCV. [Электронный ресурс]. Режим доступа: <https://kostyakulakov.ru/opencv/-обучение-каскада-хаара> (дата обращения: 19.09.2018).
2. Python API – Dlib . [Электронный ресурс]. – URL: <http://dlib.net/python/index.html> (дата обращения 20.10.2014).
3. Таненбаум Э. Компьютерные сети/ Э.Таненбаум, Д Уэзеролл. – СПб.: Питер, 2012. – 960.